Unicenter® CA-APAS® Insight Monitor for Adabas

Performance History Systems Guide

4.1

Computer Associates®

The manufacturer of this documentation is Computer Associates International, Inc.

Contents

Chapter 1: Introduction

Chapter 2: Design Overview

Chapter 3: Post-Installation and Use

Chapter 4: History File Design

Chapter 5: Processing Design

Chapter 6: Modifying History File Definition

Index

Chapter

1

Introduction

This chapter briefly discusses the purpose, approach and applicability of the Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) Performance History System. It also discusses alternative approaches to handling historical performance data produced by Unicenter CA-APAS.

Purpose

The Unicenter CA-APAS Performance History System provides a means of monitoring selected aspects of Adabas performance and workloads through extended time periods spanning many Adabas sessions. The availability of historical performance data makes possible a broader view of overall Adabas activity and trends. Retaining historical data on Adabas workloads and performance serves the following purposes:

- Allows retrospective analysis to determine what entities had major impacts on Adabas processing at specified points in time in the past
- Reflects changes in processing efficiency of the Adabas nucleus as ADARUN parameter values, application mix and operating environment factors change, providing a basis for optimizing these items
- Reflects changes in the efficiency of Adabas applications as file definitions and program logic are modified, thus providing a basis for tuning current applications and for better design of future applications
- Reflects changes in the magnitude, composition and scheduling of Adabas workloads
- Supports calculation of standard performance factors for Adabas commands for use in estimating performance of future applications
- Provides a basis for projections of future resource consumption by Adabas applications for capacity planning purposes

Approach

The Unicenter CA-APAS Performance History System consists of:

- A single Adabas file for storing various record types that reflect different aspects and time periods of past Adabas processing (possibly over multiple databases on multiple CPUs)
- Unicenter CA-APAS requests that define the types of historical data that the Data Collector produces for posting to the History file
- Various programs for capturing, storing, purging and reporting performance data

Application

This system is an optional Unicenter CA-APAS facility. It may be used in whole, in part, or not at all. If used, it may be used as provided, or it may be modified to better meet your site's requirements.

The design of this system represents just one of many possible approaches to maintaining and utilizing historical data about Adabas performance. It is not expected to fully satisfy every organization's requirements in this area.

All programs that access the History file are provided in Natural source code form to make them easy to modify locally and to promote self-sufficiency in using either the standard or a modified version of the system.

Use of some of the features and data items of the Unicenter CA-APAS Performance History System varies between versions of Adabas. These differences are noted in this document wherever applicable.

Alternatives

Alternative approaches that maintain historical data in sequential files rather than Adabas files would avoid the expense of adding and deleting large quantities of data from an Adabas file. Sequential files may be adequate where online inquiry is not needed. User-specified Unicenter CA-APAS summary information may be written to sequential data files for subsequent reference and further summarizing. The default requests used in the Unicenter CA-APAS Performance History System are examples of producing this type of output with Unicenter CA-APAS.

Chapter

2

Design Overview

This chapter describes major aspects of the design of the Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) Performance History System.

Design Objectives

The Performance History System was designed with the following objectives in mind:

- Flexibility to meet the needs of a wide range of organizations and processing situations
- Simplicity to establish, maintain and utilize
- Reasonable human and machine cost to operate

Accordingly, performance data is stored in a single Adabas file that has provision for several different record types. Use of most record types is optional. You may design and add additional record types based on the field definitions provided or new field definitions you choose to add.

Data Sources

Adabas performance data for the History file is derived from Adabas nucleus sessions. Each session is associated with a specific Adabas database and a specific CPU. Multiple databases and multiple CPUs may have Adabas sessions reflected in the History file at the same time. Nucleus sessions may be single-user or multi-user mode and may be update or read-only.

A utility that executes with the nucleus in the same region is interpreted as a single-user nucleus session.

Data enters the History file in any of the following ways:

- The Unicenter CA-APAS Data Collector can produce several types of interval summaries of Adabas nucleus session command processing data; history update programs can then add the data to the History file.
- Users may enter session statistics or make corrections via the online update facilities of the History System.
- Session statistics may be entered into the History file by user-written online or batch programs. Normally this method would be limited only to adding jobstep CPU times or other statistics developed externally from Adabas.

Data Maintenance

When adding new interval data to the History file, the History System update programs can also automatically consolidate interval data to hourly, daily, weekly and monthly levels of summarization. Online file maintenance and inquiry programs, written in Natural, are provided to make file management easier. A purge program, executable in either batch or online modes, can automatically delete the various types of history records based on age of the data compared to user-specified retention criteria.

Consolidation and purging functions are both governed by parameters that the user supplies in retention control (RC) records. It is important that you have a full understanding of the maintenance and use of RC records before you begin to install and use the History System. RC records are discussed later in this document. You should also carefully note the comments that appear in the online screens for updating RC records.

Inquiry and Reporting

The History System includes extensive online inquiry functions, such as:

- Various views of the kinds of data that exist in the History file
- Inquiries by session number
- Inquiries by CPU and DBID
- Inquiries by time periods
- Inquiries by entities such as job name, terminal-id, Natural program, CICS transaction code, etc.
- Combinations of the above, such as entities with heaviest processing loads during a time period

Users may supplement these functions by extracting data from the History file for presentation through other statistical and graphics systems.

Chapter

3

Post-Installation and Use

This chapter gives specific instructions for post-installation and ongoing use of the Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) History System components.

Post-Installation Considerations

Use of the Unicenter CA-APAS Performance History System is optional. Before you proceed with the steps described below, you should clarify the organizational objectives to be met by using this system.

The material in this chapter assumes you wish to use the system just as it is provided. If you modify the system, be sure to consider the implications of your changes on the following set-up process. In particular, you will very likely need to modify the History file definition, some or all of the History System Natural programs, and the default requests for history interval data.

Note: We strongly recommend that you install and use the History System as distributed before attempting modifications.

History File, DDM, and Programs

Specific procedures for installing these items vary depending on the operating system environment. For installation instructions, see the chapter “Performance History System Installation” in the *Unicenter CA-APAS Installation Guide*.

Local Selection Values

Inquiry programs use global variables for some of the selection criteria on prompt screens. This eliminates the need to reenter the same values for each of a series of inquiry programs. Default values for global variables such as CPU-ID and DBID can be customized for your site by editing program MENU and recataloging it. Statements to change are commented.

SORT Option

Some of the history programs contain a SORT statement whose capacity is limited by the size of Natural's FSIZE parameter. This sorting restriction can be avoided by using alternate programs which use FIND...SORTED instead of SORT. Sorting capacity of the alternate programs is restricted by the Adabas LS parameter that, unlike FSIZE, has no stated upper limit. The number of Adabas I/Os and the response time of a given alternate program could be better or worse than the original program depending on the contents of your History file.

If you want to use the alternate programs, edit program HISTSUMM to change the names of programs that it fetches. The FETCH statements and alternate program names are commented in HISTSUMM. After making the changes, catalog the modified version of HISTSUMM.

Search Criteria

Some programs provide optional methods of structuring the search criteria of FIND statements. In these cases, one option places certain criteria in the WITH clause, while the other option places them in a WHERE clause.

These options are provided for performance reasons. Comments at these points in the program explain how the options relate to conditions in a given History file.

Using the Performance History System

Requirements and techniques for ongoing use of the History System are described below. Options and requirements for adding data to the History file are discussed below.

Record Retention Criteria

One Retention Control (RC) record must be added to the History file for each combination of CPU-ID and DBID for which you plan to store historical data.

Note: This step must be performed before the update programs add records to the History file for any given combination of CPU and database.

This is also the way to control automatic consolidation of basic interval data to higher-level time periods and to control aging and purging of data records for each combination of CPU-ID and DBID. This step cannot be done until after the History file has been loaded into the database, the APAS-HISTORY DDM has been loaded into the Natural System file, and the history Natural programs have been loaded and cataloged in the Natural System file.

Adding or Revising RC Records

The procedure for adding or revising RC records is:

1. Invoke Natural.
2. LOGON APASHIST or a library-id that was substituted locally for the Unicenter CA-APAS Performance History System.
3. Execute MENU and select the update functions option.
4. Select X UPDATE FUNCTIONS FOR HISTORY FILE
5. Select R RETENTION CONTROL (RC) RECORD MAINTENANCE
6. Use this option to add or modify a retention control record for each combination of CPU-ID and DBID and to specify retention periods for the various record types that may exist for each such combination.

Retention Period

Retention periods are specified in terms of months and days, in the format MMDD. This is NOT a date. It is simply a length of time from the date within a record, and not necessarily the date the record was added to the History file, to retain records. A value of 9999 means that no automatic purging is done for the record type or level; retention is indefinite.

A zero-value retention period for a given record type or consolidation level means:

- No records of that type are added to the History file by the history update programs.
- If executed, APASHPRG deletes any records of that type regardless of the dates within the records.

A zero-value retention period for a consolidation level (hour, day, week or month) also means that no automatic consolidation of interval data to that level occurs for any ‘Nx’ type records.

If the interval specified in the history SUMMARIZE requests is greater than 30 minutes, then a zero-value retention period should be specified for the hourly consolidation level.

Retention periods may be revised anytime you wish; simply use the same online facility to change retention periods.

Session Statistics Data

Session statistic information is processed by using the Unicenter CA-APAS requests HUP5ACT, and optionally HUP5PRM, and history update programs APASHUP5 and APASHUP2.

Command Processing Data

The History System includes Unicenter CA-APAS requests that produce output files containing records reflecting summaries of Adabas command processing for specified regular time intervals. The interval data in output files is posted the to the History file by history update programs. The following types of interval data are supported:

- ECB count frequencies—request ECBSUM, update program APASHUPE, JCL HUPEJCL.
- All commands by command code—request ALLCMDS, update program APASHUPA, JCL HUPAJCL.
- Commands by CICS trancode—request TRANCMDS, update program APASHUPC, JCL HUPCJCL.
- Commands by file—request FILCMDS, update program APASHUPF, JCL HUPFJCL.
- Commands by jobname—request JOBCMDS, update program APASHUPJ, JCL HUPJJCL.
- Commands by Natural program-id—request NATCMDS, update program APASHUPN, JCL HUPNJCL.
- Commands by terminal-id—request TRMCMDS, update program APASHUPT, JCL HUPTJCL.

Interval Values

Each of the requests specifies an interval value. An interval is mandatory, but you may use an interval value that is different from the one supplied in the default requests.

The default interval value is 15 minutes. You may specify any interval that divides evenly into 24 hours. Of these possibilities, the following are the most reasonable:

- Fifteen or 30 minutes
- One, 2, 4, 6, or 8 hours

If an interval greater than 30 minutes is used, then you should set the retention period for hourly consolidation to zero.

Note: Before deciding to use any of these types of interval data, consider the quantities of records that are generated in the History file to avoid producing more records than you care to process and store.

Estimating Record Quantities

One key factor influencing the quantity of records generated is the length of interval specified in the request. The longer the interval specified the fewer records generated. The following guidelines should be helpful in estimating record quantities (15-minute intervals are assumed):

- ECBSUM—one NE record per interval. Maximum of 96 records per day per Adabas session.
- ALLCMDS--one NA record per command code used per interval. Maximum of approximately 120 records per hour or 2880 records per day, per Adabas session. One CA record per period; maximum 1 per month, 1 per week, 1 per day and 1 per hour.
- TRANCMDS--one NC record per CICS transaction code used per interval. Maximum depends on how many CICS transaction codes are defined and used at your site. One CC record per period, per CICS transaction code.
- FILCMDS—one NF record per file number referenced per interval. Maximum of about 1,020 per hour or 24,480 per day per Adabas session. One CF record per file number per period; maximum 65535 per month, 65535 per week, 65535 per day and 65535 per hour.
- JOBCMDS—one NJ record per unique job name per interval. One CJ record per job name per period.
- NATCMDS—one NN record per Natural program-id per interval. One CN record per Natural program-ID per period.
- TRMCMDS—one NT record per terminal-id per interval. One CT record per terminal-ID per period.

Processing Interval Data

The following steps are required to process interval data:

1. Decide what types of Command Log related records you want to maintain in the History file.
2. Use default requests from member APSDFLT3 in the Unicenter CA-APAS source library to have one or more interval summary output files generated. (If you change the requests, then change the update programs that post the interval data to the History file.)
3. If you run multiple concurrent Adabas nucleus sessions on a given CPU for a given database, use different CPU-ID values on the Unicenter CA-APAS GLOBALS statements.

4. Arrange to have the requests processed by the Data Collector (in either batch or MPM mode) for each Adabas nucleus session (single or multi-user) of interest. Establish an appropriate disposition of the file(s) into which the history records are written. You can accumulate records for multiple sessions within a single file before running the update programs to post the interval data to the History file.
5. Arrange for the appropriate Unicenter CA-APAS update program(s) to be run periodically to process the output file(s) produced by the Data Collector.

Points of Consideration

When running the Unicenter CA-APAS history update programs, the following points should be considered:

- Whether they are submitted automatically (such as once during each Adabas multi-user session) or are submitted manually (such as once per day, once per week, etc.) to process the history data accumulated from previous sessions.
- Whether the Unicenter CA-APAS programs are run in single or multi-user mode.
- What kind of recovery provisions are appropriate to assure proper contents of the History file if a run fails to complete successfully. Each of the programs contains logic for automatic restart capability; this logic should be reviewed for compatibility with your operations and procedures.
- The programs that post data to the History file terminate if they detect that records about to be added already exist on the History file. This is to prevent duplicate data in the History file in cases where input files are erroneously processed more than once.

In the case of the programs that process interval data, the check for duplicate data is effective only if the retention period for a given type of interval data record is greater than zero. In cases where the retention period for interval data records is zero but the retention period for consolidated data records is greater than zero, it is still possible to duplicate the addition of interval data to the consolidated data records.

- Each program that posts interval data to the History file continues processing if it encounters input data records other than its own type. This creates the option of having the history requests share a common OUTPUT-FILE (either single or dual). The benefit of using a shared file is simplicity. There are fewer files to allocate, to define in JCL and to manage. The drawback is that all of the records in the shared file must be read by each of the update programs that are being used at a given site.
- It is preferable to execute APASHUP5 before execution of any of the other update programs.

Data from Non-Unicenter CA-APAS Sources

Some non-Unicenter CA-APAS sources of Adabas related performance data could include:

- CPU-monitor
- I/O monitor
- Job step accounting
- TP monitor processing statistics
- Application generated statistics

If data from such sources are to be maintained in the Unicenter CA-APAS Performance History file, then the following steps must be carried out:

1. Determine what data sources are used, what data items are captured from each and how the data is related to other kinds of data in the History file.
2. Make necessary modifications or additions to the History file field definitions to provide for appropriate History file record types and data fields to accommodate the data from the non-Unicenter CA-APAS sources.
3. Provide mechanisms to capture the additional data and post it to records within the History file. This might be as simple as someone reading step accounting information after each Adabas session and using an online Natural program to post measured step CPU time to the NB or UB type record for the session, or it might involve the creation of additional JCL and user-written programs.
4. Provide mechanisms to age and purge any additional record types that have been created.
5. Provide programs to report the additional record types and/or fields.

Restart Procedures

Most of the history update programs are designed to be restartable following abnormal terminations such as:

- The Adabas session supporting execution of a history update program is interrupted.
- Execution of a history update program is interrupted.
- A history update program terminates with explanatory messages due to detected error condition(s).

The restart logic uses the Adabas transaction recovery facilities; including the use of application-supplied restart data which Adabas stores in its checkpoint file as part of processing ET commands.

Understanding and Using Restart Capabilities

The following points are provided to assist you in understanding and using the restart capability:

- Each history update program that is driven by an input file from a Unicenter CA-APAS history request defines logical transactions of work that are related to record counts on the input file.
- At the conclusion of each logical transaction, the history update program uses an END TRANSACTION statement to have Adabas post restart data to the Adabas checkpoint file. This restart data includes an input file record count from which a restart could be started. It also includes an identifier for the program.
- When a history update program completes normally, it uses an END TRANSACTION statement to send null restart data to the checkpoint file. This data indicates that no restart is needed.
- When executed without any execution parameters (example: EX APASHUP5), a history update program queries the checkpoint file to see if it needs to restart and continue processing an input file that was not finished during the previous execution of the program. If a restart condition is indicated, then the program reads the current input file, attempting to locate and confirm the restart position on the input file. If location of the restart position is successful, then normal processing resumes from that point. Otherwise, the execution is terminated. In either case, informative messages regarding the restart effort are given.

- It is the user's responsibility to monitor messages from each execution of each history update program and do one of two things each time a history program fails to complete normally:
  - Ideally, the failed history update program should be executed again (before any other history update program executes) with the same input file and with no execution parameters.
  - However, if the same input file is not available, then the next execution of the program should contain the NORESTART parameter to suppress the restart logic.

    ```
    APASHUP5 NORESTART
    ```

- Adabas stores restart data in the checkpoint file under user-id values that are supplied by the application programs. Natural provides these id values for the history update program. If several history update programs are executed from within a single batch execution of Natural, then Natural uses the same id (*ETID) for all of them. In this case, each successive history update program overlays (deletes) the restart data left in the checkpoint file by the previous history update programs.

  To have history update programs use different *ETID values from one another, execute each program in a separate batch Natural job which uses a job name different from those used with any of the other history update programs.

Maintaining a Session-Id Control Record

A session-id control record is automatically created the first time one of the update programs is executed. This record should normally require no user attention.

However, if problems with the session-id control record arise, there is an online function that resolves most of them. (See the source code for details.) The steps to invoke this function are as follows:

1. Invoke Natural.
2. LOGON APASHIST (or other local library-id).
3. Execute MENU and select X Update Functions For History File.
4. From the Update Functions Menu, select S Session Control (SC) Record Maintenance option.

The program previews its actions; select the option as appropriate.

Chapter

4

History File Design

This chapter describes the logical and physical views of the Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) History file.

Structure Overview

To appropriately update the History file and to correctly interpret information derived from it, you must first understand the logical and physical structure of the History file.

The APAS-HISTORY DDM (Natural Data Definition Module) and the Adabas file definition, obtained from the field definition statements or an ADAREP report for the file, should be consulted for detailed definitions of the individual fields in the Performance History file. A cross reference of the fields and the record types in which they are used is provided in Unicenter CA-APAS source library member HISXRF.

You may find it helpful to glance at the field definitions before reading further. Then, as you read about each kind of record, you may find it useful to refer again to the field/record cross reference for more insight into the specific contents of the record. To facilitate these references, we suggest that you now print the HISXRF member from the source library and the APAS-HISTORY DDM from the Unicenter CA-APAS History Natural library before reading further in this section.

History File Record Types

History file record types, along with relationships between certain record types, are described below.

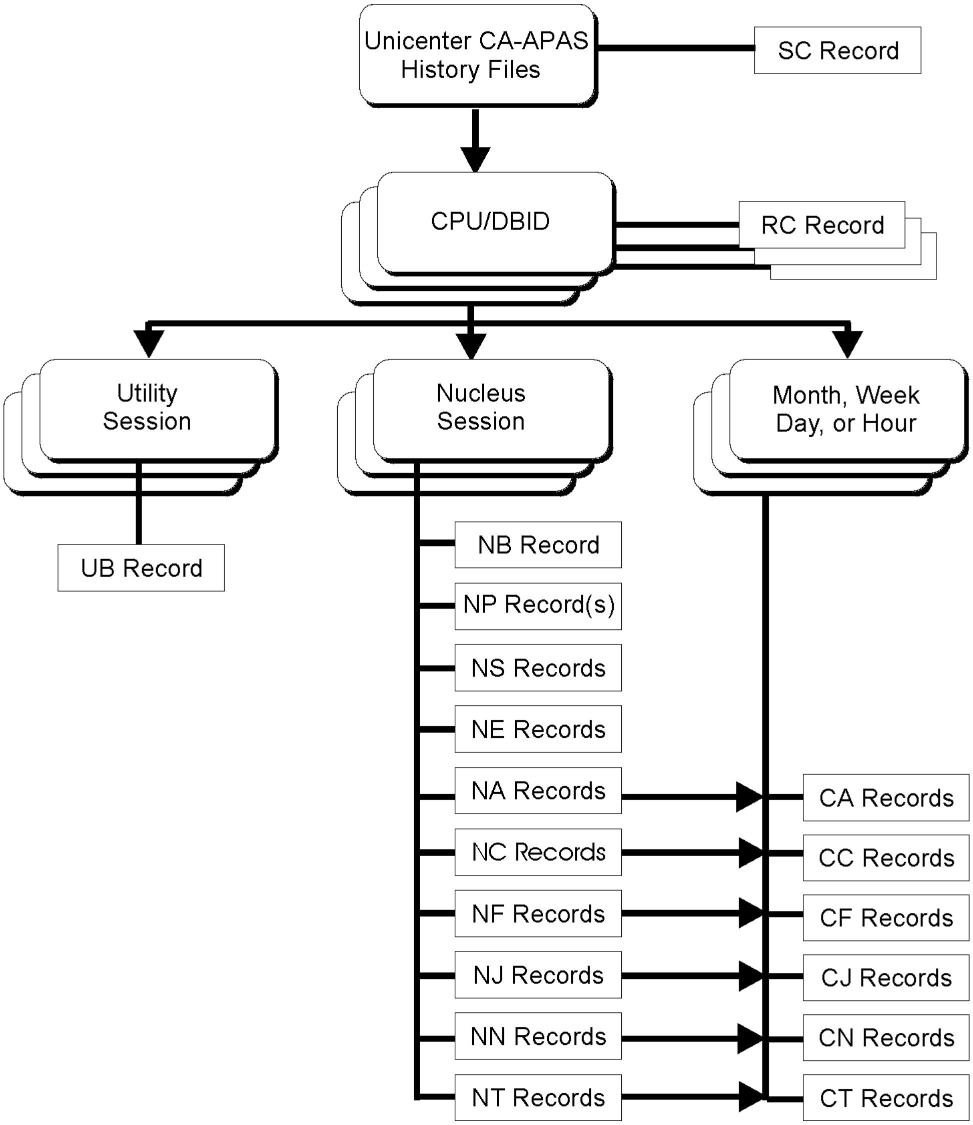
Record Type	Description
SC	Session Control (Session-Id Number Assignment).
RC	Retention Control Parameters.
UB	Utility Session Basic Statistics.
NB	Nucleus Session Basic Statistics.
NP	Nucleus DPARM Values.
NS	Nucleus DSTAT Statistics.
NE	Interval ECB Count Frequencies.
NA	Interval All Commands Summary.
NC	Interval CICS Trancode Commands Summary.
NF	Interval File Commands Summary.
NJ	Interval Jobname Commands Summary.
NN	Interval Natural Pgm-Id Commands Summary.
NT	Interval Terminal-Id Commands Summary.
CA	Consolidated All Commands Summary.
CC	Consolidated CICS Trancode Commands Summary.
CF	Consolidated File Commands Summary.
CJ	Consolidated Jobname Commands Summary.
CN	Consolidated Natural Pgm-Id Commands Summary.
CT	Consolidated Terminal-Id Commands Summary.

Each of these record types is described in detail later.

History File Logical Structure

The logical structure of information within the History file is shown below.

History File Logical Structure



Historical data are associated with a particular CPU/database combination. The different record types reflect what happened during an entire nucleus session, during a defined interval within a nucleus session, or during a defined time period that could span multiple nucleus sessions.

Control Records

The record types used to control the programs that maintain the History file are described below. These records contain no data about Adabas performance.

Session-Id Control Record (SC)

One record of this type is maintained in the History file. It contains the most recently assigned session-id number. Each time a new UB or NB record is added to the file, the SC record is read and updated in order to increment the control session-id number.

The update programs that add interval data to the History file attempt to associate each interval's data with a particular nucleus session. This is done by searching for an existing NB record whose session start and end date/time stamps span the interval. If such an association is not possible, the update programs use the next available session number from the SC record as the session-id for the interval records.

A special program, APASHUP2, is provided for execution immediately following the program APASHUP5 that creates NB records. APASHUP2 attempts to automatically associate any such “orphan” interval records with recently added NB records. In addition, an online function is provided to allow users to directly change the session numbers in interval records. Either of these methods allows interval records to be associated with an NB record which was created subsequent to the interval records being added to the file, should this occur.

Retention Control Records (RC)

For each combination of CPU-ID and DBID, one of these records controls the following maintenance functions:

- Which record types may be added to the History file
- Which levels of automatic consolidation of interval data to hour, day, week and month levels may be done
- How long each record type is retained in the History file

Note: A single record of this type must be present in the History file for each combination of CPU-ID and DBID before any data about that CPU-ID/DBID combination can be added to the History file.

An online function for creating and revising RC records is provided. This function is discussed in more detail in see the chapter “Post-Installation and Use.”

Session-Wide Statistics Records

Each execution of the Adabas nucleus constitutes a distinct Adabas session.

None of the record types reflecting session-wide statistics are ever broken down or consolidated upward to different time periods. Once present in the History file, you may add information to these records from other sources using user-written programs or with Unicenter CA-APAS online programs.

All session statistics are produced by Unicenter CA-APAS requests.

Nucleus Session Basic Statistics Records (NB)

Information about the quantity and efficiency of command processing for nucleus sessions is of high value in performance management. Each nucleus session should be reflected in the History file by a single record of this type. This type of data is produced by Unicenter CA-APAS requests.

Nucleus Session Parameter Records (NP)

A Unicenter CA-APAS request can produce selected nucleus parameters.

Storage of Adabas parameter values for each nucleus session may or may not seem worthwhile at your site. In general, it's probably better to go ahead and record it in case the information is later desired; the storage cost and bother are insignificant. In particular, one specific parameter, LBP, may by itself justify the effort since the actual size of the Adabas I/O buffer, rather than just the size requested, is recorded.

Some nucleus parameters are not recorded.

Interval Statistics for Nucleus Sessions

Two different approaches to storing interval statistics are provided. The two kinds of information are complementary rather than redundant.

One approach stores high-level summary statistics for regular intervals of the session (in NS records, described below). These statistics are not broken down by Adabas file, user, etc. They reflect overall loads and efficiency during each interval. This category of statistics is produced by a Unicenter CA-APAS request.

The other approach uses Unicenter CA-APAS summaries of nucleus command processing data for defined intervals during nucleus sessions, broken down by related entities such as command code, file numbers, job names, terminal-ids, CICS transaction codes, or Natural programs.

Nucleus Statistics Records (NS)

The longer a nucleus session is and the more variable the workload throughout the session, the less meaningful some of the session total statistics become. NS records allow nucleus processing activity during specified intervals of a nucleus session to be recorded. The intervals may then be analyzed independent from one another and can be related to other types of information applicable to the individual intervals. One major use of this finer information is to reflect the impact of individual batch jobs on nucleus format translation, I/O buffering and thread utilization.

Derivation of NS Records

Creation of NS records is optional, under control of the retention control period parameters for the NS record type specified by the user in RC records. Information from this record type is never consolidated upward to higher-level time periods.

Data for NS records is developed by the HUP5ACT Unicenter CA-APAS request. The output from this request is posted to the History file by History file update program, APASHUP5.

Interval ECB Count Frequency Records (NE)

A single NE record for each interval carries within the occurrences of a periodic group the frequency of each ECB count that actually occurred during the interval. Each occurrence number of the periodic group represents the ECB count of the same value--with the exception that occurrence 99 represents all ECB counts greater than 98, an unlikely eventuality. The contents of the single field within the periodic group is the number of times the particular ECB count occurred during the interval.

The maximum ECB count during an interval is carried as a field value outside the periodic group for easier access in certain kinds of reporting. This information can be useful in reflecting patterns of relative saturation of Adabas throughput capacity during a given interval and across a series of intervals.

It loses meaning as the time span increases, so this type of interval data is never consolidated to higher-level time periods.

ECB counts tend to be of value only in analyzing saturation in given instances, therefore it should normally be unnecessary to collect or store this type of information until saturation problems have occurred or are suspected. The retention period for any of these records should normally be very short.

Interval Command Processing Summary Records

Each record of the types in this category reflects the quantity and selected processing characteristics for commands during a given interval. The different record types allow for recording statistics that were summarized by individual entitles, such as jobname, terminal-id, or file number, that are associated with particular sets of commands.

The initial usefulness of this short interval data is in what it reveals about workload and performance within the given interval. Next, it shows the relative workloads across successive intervals within a given day. Then it provides a basis for comparison of the same intervals across successive days.

Another important reason to acquire interval data is that it is the means of developing summary data for hours-of-days, days-of-months, weeks-of-years and months-of-years. Automatic consolidation is controlled through a retention control record for a particular CPU-ID and DBID combination.

Once interval records have been consolidated upward to these higher-level time periods, they could be purged if they are no longer needed at the original interval level. A short retention period would avoid the accumulation of an excessive number of interval level records; this is particularly significant for NC, NF, NJ, NN and NT record types unless output has been restricted to only a limited number of selected transaction codes, files, jobs, Natural programs or terminal-ids.

By Command Code (NA)

Each of these records summarizes all activity for a particular command code during an interval. This can be useful in showing the relative mix of command codes and which kinds of commands may have been causing excessive loads or encountering bottlenecks (enqueuing of complex searches).

By CICS Trancode (NC)

Each of these records summarizes all command activity for a CICS transaction code during an interval. This can be useful in relating Adabas processing to CICS statistics for the transaction codes.

By File Number (NF)

Each of these records summarizes all command activity for an Adabas file number during an interval. Activity by file number is usually one of the most useful breakdowns to use as a starting point in analyzing Adabas workloads and performance. Files that are heavily used or that have suspicious command mixes or characteristics typically indicate a need to look at the programs that access the files.

By Jobname (NJ)

Each record summarizes all command activity for a job name within an interval. Multiple submissions of a given jobname may be reflected within a single record.

The standard request reflects all job names. This allows the total Adabas processing done from each copy of teleprocessing monitors to be reflected along with batch processing. If only batch processing is desired, a WHERE clause could be added to the JOBCMDS request to exclude TP and/or TSO user types.

By Natural Module Id (NN)

Each of these records summarizes all command activity for a Natural module id during an interval. Note that the id includes the Natural system file number, library name, and module type as well as the program name.

By Terminal-Id (NT)

Each record of this type summarizes all command activity for a terminal-id during an interval.

Note that the TRMCMDS request, as distributed, includes each batch job step as an additional 'terminal'. This request should definitely be modified to exclude batch jobs. The appropriate WHERE clause condition depends on which Unicenter CA-APAS options are being used at a given site. See the comments in the request for suggestions.

Consolidated Command-Processing Records

Any of the interval-level command processing data record types described earlier may be automatically rolled up to related record types within this category for standard hour, day, week and month periods. The basic interval-level data are consolidated across sessions to reflect total processing loads in terms of total number of commands, total number of I/Os to each of the significant Adabas data sets, and total CPU time.

The time period that a given record in this category reflects is carried in the field PERIOD that contains a composite value depending on the level of the period.

The structure of the PERIOD field is:

Level	Level Code	Year	Week or Month	Day	Hour
Month	M	**yyyy**	**mm**	**–**	**–**
Week	W	**yyyy**	**ww**	**–**	**–**
Day	D	**yyyy**	**mm**	**dd**	**–**
Hour	H	**yyyy**	**mm**	**dd**	**hh**

Special superdescriptors have been defined for each of the Cx type records to facilitate logical sequential processing for the types of reporting that are anticipated for each record type. These keys are described below in the discussion of each record type.

All Commands (CA)

A major benefit of consolidating volumes and overall characteristics of all commands over longer time periods would be the ability to correlate changes in loads with introductions of new applications and to correlate changes in efficiencies with changes in resources, configurations, or parameters. Also, showing patterns of loading could be useful for scheduling and resource planning.

Special keys defined for use with the CA records are CA-TIMESEQ-KEY for developing total Adabas workloads on all CPUs and databases across time periods, and CA-SYSTEM-KEY for developing workloads for individual CPUs across time periods.

By CICS Trancode (CC)

These records have much the same kinds of uses as at the interval level except for the change in time spans. Special keys provided for the CC records are CC-TIMESEQ-KEY and CC-TRAN-KEY.

By File Number (CF)

Consolidated command processing volumes and characteristics for individual files has much the same kinds of uses as at the interval level except for the change in time spans. Special keys provided for the CF records are CF-TIMESEQ-KEY and CF-FNR-KEY.

By Jobname (CJ)

This type of data could be used to identify heavy volume batch programs that run relatively frequently and might merit review for performance tuning. Special keys for the CJ records are CJ-TIMESEQ-KEY and CJ-JOB-KEY.

By Natural Module Id (CN)

One possible use of this type of data would be to rank Natural programs according to frequency of usage as reflected by the number of Adabas calls. This would identify those programs which should be evaluated for possible conversion to more efficient languages. Special keys for the CN records are CN-TIMESEQ-KEY and CN-NAT-KEY.

By Terminal-Id (CT)

This type of data could reflect trends in Adabas usage from a network of terminals, indicating the number of terminals active and workloads from the terminals. Special keys for the CT records are CT-TIMESEQ-KEY and CT-TERMINAL-KEY.

Chapter

5

Processing Design

This chapter describes the processing capabilities that exist in the Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) History System as it is delivered. Some of these capabilities are optional and therefore need not be used at a given site.

History File Maintenance

Maintenance of the History file is largely a matter of making decisions as to what kinds of historical information are desired and then setting up continuing procedures for the generation, capture and routing of inputs to the file. For the major data flows and processing steps for History file maintenance, see Adabas Processing in this chapter.

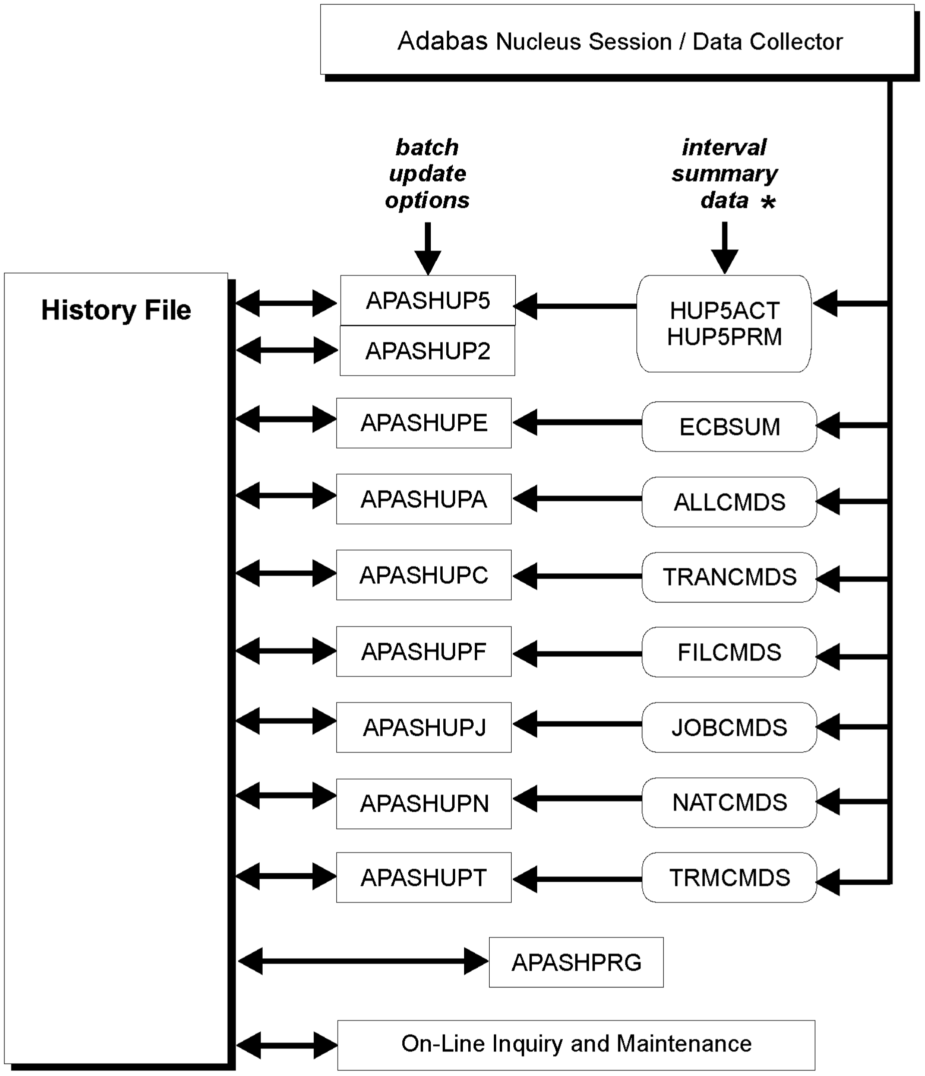
Little, if any, manual updating of information in the History file should be required. Purging can be handled automatically. Batch programs are provided for the routine processes of adding new data to the History file and consolidating interval-level data up to broader time periods.

The Unicenter CA-APAS online menu provides a number of selections for infrequent maintenance functions associated with assigning session-id numbers and setting retention control parameters. The online selections also provide a means of entering measured jobstep CPU times for nucleus or utility sessions if desired.

Adabas Processing

The following diagram displays the History file maintenance flow in Adabas.

Adabas History Files Maintenance Flow



* The Unicenter CA-APAS history requests may all direct their output to a single common physical file that you can use as input to the various batch update programs.

Adding Session Data to the History File

Session data may come from Unicenter CA-APAS requests or other sources provided by users. All records derived from a particular nucleus session must be related to that session for accurate reporting to occur. Since some of these potential data sources are uncoordinated, user involvement in associating data from multiple sources may sometimes be required. Unicenter CA-APAS attempts to eliminate or minimize this kind of user effort however, as explained below.

Session Data

The Unicenter CA-APAS requests, HUP5ACT and HUP5PRM, produce data for NB, NS and NP records. The output from these requests is read by history update program, APASHUP5, for posting to the History file.

Command Related Data

A set of default Unicenter CA-APAS SUMMARIZE requests cause the Data Collector to generate selected sets of command processing data for specified time intervals during nucleus sessions. Each request has an associated update program that reads the output file from the request and posts the data to the History file. The requests and programs are:

Request	Program
ALLCMDS	APASHUPA
ECBSUM	APASHUPE
FILCMDS	APASHUPF
JOBCMDS	APASHUPJ
NATCMDS	APASHUPN
TRANCMDS	APASHUPC
TRMCMDS	APASHUPT

These update programs try to match each record of interval data with a nucleus session that is already reflected in the History file by an NB record. This is done by searching for an NB record whose session starting and ending date/time stamps span the given time interval. In cases where such an assumption is not logical, these programs use the Session Control record to arbitrarily assign different session numbers to the different intervals from a single nucleus session.

This method of associating Command Log data with related session data makes it highly desirable to execute APASHUP5 before execution of the update programs that process interval data from the requests shown above. This sequence of program executions results in the NB record for a given session already being present in the History file when interval data from the same session is being processed. Where this sequence of program executions is not possible, execution of APASHUP2 immediately following each execution of APASHUP5 should result in reconciliation of records for most sessions to a common session-id number for each session. Additionally, online functions are available to allow manual corrections when necessary.

In each execution, the update programs log all assignments of session numbers. You should review this information to determine if any manual adjustments are necessary.

Consolidating Data to Higher Time Periods

Consolidation of data from the original intervals to higher-level time periods is done automatically by the update programs according to parameters in retention control (RC) records.

When a new interval record is added to the History file, data from the new record is added to all appropriate higher-level time periods if a retention control record for the CPU-ID and DBID combination involved exists and specifies a non-zero retention period for a given level of data. Creation and maintenance of retention control records may be done through online Unicenter CA-APAS programs.

Aging and Purging the History File

Purging of records from the History file may be done automatically by executing APASHPRG as a step in the batch update cycle or from an online menu.

APASHPRG's purging action is governed by parameters in retention control records for the various CPU-ID and DBID combinations. Purging takes place if the difference between the current date and the date of information within a record equals or exceeds the specified retention period obtained from an RC record.

Note: Each retention period you specify is simply some number of months and/or days (from the date of a given record) to retain a record in the History file; the retention period is NOT a date.

Inquiry and Reporting

A number of Natural programs are provided for History file online inquiry and reporting. They are available to be selected from the History System main menu program, MENU, under the library-id APASHIST. Some of the functions included are:

- History file contents inquiry
- Utility session reporting
- Nucleus session basic reporting
- Nucleus session interval command processing reporting
- Nucleus session ECB reporting
- Reporting of consolidated command processing data for hour, day, week and month time periods

The online inquiry and reporting programs are useful as distributed, and they also provide a convenient starting point for you to develop many of the additional types of inquiries and reports you may want.

Chapter

6

Modifying History File Definition

This chapter discusses possibilities for locally developed changes or extensions to the standard Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) History System. Since History System components are provided in source form, there are few restrictions on the ways in which the system may be modified.

It is important to note that most modifications to the History file definition require corresponding changes in the Natural programs and the APAS-HISTORY DDM that are distributed with the History System.

Deleting Fields

One initial type of modification to consider is whether to delete certain fields from the file. Some possibilities might include:

- Some of the Adabas parameters that have no significant bearing on performance
- DBNAME if DBID is sufficient and convenient
- Any fields unique to record types that are never used
- Any descriptors that are never used
- Any fields not of interest to the users

Any programs that reference deleted fields must be modified and re-cataloged.

Revising Fields

If different names are desired for any of the standard fields, the new names should be defined as synonyms to avoid having to change field names in the programs.

Additional revisions might include changing, adding or deleting field headings or edit masks. Changes to headings or edit masks may require modification and re-cataloging of programs that use the fields involved.

Defining New Fields

Field definitions may be added to the file to provide for storing information from non-Adabas sources. These might include I/O or CPU-monitor statistics for intervals that match Adabas interval data. Adabas field names IA through J9 are reserved for local use, and are not used when adding new field definitions to the History file.

Defining New Record Types

Record type codes UA through U9 are reserved for local use; they are not used when defining new record types for the History file.

Additional Inquiry and Reporting

The online displays have been limited in content to screen width; existing WRITE and DISPLAY statements could be extended with additional output elements if the programs were run in batch mode. Also, there is considerable opportunity to tailor the selection logic in most of the programs to more precisely target the information you wish to see.

A major direction for augmenting reporting from Adabas performance data in the History file is that of extracting data for further processing and more sophisticated forms of graphic presentation.

Natural provides an excellent method of extracting information and writing it to a sequential work file for subsequent input to other software packages.

Index